[Reprinted from the Journal of the American Chemical Society, 97, 5755 (1975).]

Applications of Artificial Intelligence for Chemical Inference. XVII. An Approach to Computer-Assisted Elucidation of Molecular Structure[1,2]

Raymond E. Carhart,* Dennis H. Smith, Harold Brown, and Carl Djerassi

Contribution from the Departments of Chemistry and Computer Science, Stanford University, Stanford, California, 94305. *Received March 27, 1975*

Abstract: An interactive computer program for assisting in molecular structure elucidation is described. The program is supplied with information in the form of inferred structural fragments of an unknown together with a variety of constraints on desired and undesired structural features. The program generates all structural isomers, without duplication, consistent with this information. Our approach employs a method of atom and superatom assembly in which superatoms are imbedded within intermediate structures to yield final structures. This method permits a stepwise solution of a problem during which intermediate results can be examined interactively and constrained further during the course of generation of final structures. The program suggests solutions to a structure problem and provides a guarantee that no plausible alternatives have been overlooked.

Automation of tasks related to elucidation of molecular structure has been the focus of many, diverse research groups. Automation of analytical instrumentation, e.g., NMR spectrometers, X-ray diffractometers, represents one area of effort. Sophisticated computer programs for analysis of data in terms of molecular structure represent another broad area. This report is concerned with the latter area and describes a computer program for assisting in molecular structure elucidation based on structural features of unknown molecules derived from physical, chemical, and/or spectroscopic information.

Our program is designed to model some aspects of manual approaches to structure elucidation. These manual approaches normally involve piecing together structural fragments of arbitrary complexity, inferred from a variety of sources of information. As structures are constructed in this way, chemical knowledge and intuition serve to constrain the structural types considered plausible. Knowledge of the sample results in early elimination of unstable species, unlikely functional groupings, and so forth. Knowledge of symmetry helps prevent consideration of equivalent (duplicate) structures. We know that people well versed in the "art" of structure elucidation are capable of making intuitive leaps from data to plausible structures with surprising accuracy. Such leaps rely on broad chemical experience, reasoning by analogy, and intelligent guessing, none of which can easily be modeled in current computer programs. The task of assembly of inferred structural units into complete structures, however, is amenable to systematic treatment, as we have demonstrated for assemblies of atoms without constraints[3b] and which we discuss in this report for assemblies of structural units of arbitrary complexity (may be atoms) under constraints.

Our program, which we call CONGEN (for *CON*strained structure GENeration), represents the next step in our continuing efforts directed toward application of artificial intelligence[4] techniques to the area of chemical inference associated with structure elucidation.[3] It is important to set the context of CONGEN within molecular structure elucidation and illustrate both what it is designed to do and what it will not do.

CONGEN's computational model is heuristic search.[5] We have found this a useful model for representing problem solving in this area of chemistry. This model is also implicit in programs written for other areas of chemistry, such as computer-assisted organic synthesis.[6] Heuristic search assumes the ability, at least in principle, to determine all possible solutions to, or next steps to be taken in, a problem. We do not claim that manual approaches to structure elucidation necessarily use heuristic search. As mentioned above, intuitive leaps frequently bypass more systematic considerations of possibilities. It is part of the challenge to a computer program that it have the flexibility to allow the user to exercise his intuitions and circumvent undesired pathways to solutions.

Heuristic search is frequently implemented in three phases, referred to as PLAN, GENERATE, and TEST. During PLAN, available information, e.g., data, rules of interpretation, are examined to infer constraints to be used in the GENERATE phase, where the generator is the proposer of solutions within these constraints. The results of generation are then evaluated during a TEST phase, where any additional information is brought to bear on candidate solutions to attempt to restrict them further.

CONGEN is designed primarily to perform the GENERATE phase of PLAN, GENERATE, and TEST, under constraints supplied by the user. The complex, frequently ill-defined process of inference of structural fragments and constraints, i.e., planning, is left to the chemist. CONGEN performs some internal planning as it determines the best place to implement a given constraint. But although the program can use structural information determined by automatic analysis of data (e.g., ref 3a), CONGEN by itself performs no such analysis.

The program also assists in the TEST phase as it provides a mechanism for implementing additional constraints imposed on existing candidate structures. Thus, CONGEN, although related to the work of others,[7-9] is focused in a different way on the overall problem of structure elucidation. That part of the problem which admits of formal mathematical treatment (and which is most difficult for chemists to do exhaustively) is given to CONGEN.

Other laboratories are developing programs, based on different computational techniques, for application to problems similar to those addressed by our program. Sasaki and coworkers[7] have described ambitious efforts at automation of the complete task of structure elucidation, from initial acquisition of data through to proposed structures. Their program is capable of exhaustive generation of structural isomers from a given empirical formula.[7c] The computational technique involves calculation of a canonical representation for each isomer.[7d] The technique can only be constrained by supplying structural fragments rather than atoms. Because the list of structural fragments which can be used is small, and because other important constraints (see Method section) cannot be used, the program has limited generality.

Munk and coworkers[8] have discussed a program which has aims similar to those of our program. Their program is designed primarily for problems in which most of the atoms are included in polyatomic fragments. Duplication can become a severe problem when many equivalent fragments (e.g., single atoms) must be used to construct structures. Their program is capable of handling several types of constraints to restrict the generation of undesired structures. Although more limited than our approach (in terms of flexibility, user interaction, and avoidance of duplicate structures), application of the criterion of production of useful results would rate the program a decided success.

Gribov et al.[9] have published descriptions of a computer-based system designed to examine spectroscopic data for structural features of molecules and to test, by calculation of expected spectroscopic behavior, candidate structures. In this program, however, no attempt is made to generate structures automatically; this part of the procedure is done manually.

We feel that CONGEN provides a more general and flexible approach than other related attempts[7-9] to the use of computational techniques for structure elucidation problems based on physical, chemical, and spectroscopic data. The program allows a problem to be stated and constrained using a language of structural fragments. The constraints available are those which are normally brought to bear in manual approaches to piecing together this information. It provides the user with the tools to solve such problems in a systematic, thorough way.

Method

Our approach to computer-assisted structure elucidation employs two major, distinctive features—the actual method of structure generation and user interaction with the program. Structure generation utilizing assemblies of atoms and inferred structural fragments ("superatoms"[3b]) employs a technique known as "imbedding." Intermediate structures are first generated using only the names of superatoms (see **9–13**, **14–18**, below). Each intermediate structure may represent a whole class of final structures. By giving the user access to the problem at this level, we provide the capability to eliminate large numbers of final structures by removing ("pruning") a small number of intermediate structures. Imbedding refers to the procedure whereby superatom names are subsequently expanded into their full identities. This technique, discussed in detail below, facilitates solution of complex problems by permitting stepwise assembly of structures under constraints. We feel that this method reflects some aspects of the strategies used by chemists in piecing together structural information, although in the absence of a systematic procedure, no two people will use exactly the same method.

The ability to guide this procedure interactively to solution(s) helps prevent unmanageable combinatorial explosions, i.e., construction of vast numbers of undesired structures. Examination of intermediate structures frequently suggests additional constraints which were overlooked previously. Applied at the point of discovery, they reduce the problem before the next step is taken.

We intentionally avoid discussions of programming details. Note, however, that the mechanisms used within the program to constrain the structure generation process are an intricate mix of computational techniques peculiar to the process of structure generation[3b,c] and automation of some strategies used by people in solving these problems. The more interesting of these strategies are indicated in the subsequent discussion. Additional information on the method is available (see Experimental Section).

The method is outlined in Figure 1. In general there are many ways to approach a given structure problem using the program. The scenario in Figure 1 is a functional description and represents only the general flow of a typical CONGEN session. At each level, the user can examine the current structures and define and implement new constraints if desired.

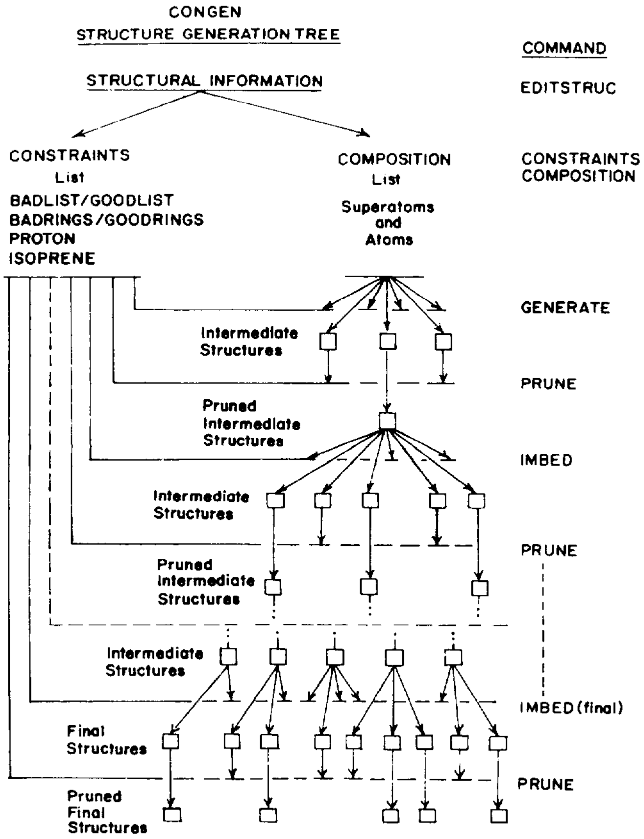
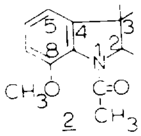
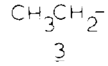
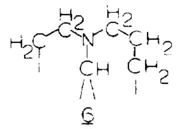


Figure 1. A functional diagram of CONGEN. Structural information supplied to CONGEN via the structure editor EDITSTRUC is used on a COMPOSITION list of atoms and superatoms which are used to GENERATE structures under the control of the CONSTRAINTS list. Intermediate structures which obey the constraints can be further restricted using PRUNE. A stepwise process of restoration of superatoms to their full identities (IMBED) eventually yields final structures. Existing or additional constraints can be implemented at each step in this procedure, including further use of PRUNE on final structures based on new data.

At any point in the method, a number of auxiliary functions not mentioned in Figure 1 can be invoked. For example, the user can draw some or all of the current structures, save results for later use, restore previous results, restart a problem, exit the current command, and so forth. These are not discussed further as they deal more with the mechanics of using the program than with a presentation of the approach. However, these auxiliary functions are an absolutely essential part of any useful interactive system, and we do not want to neglect completely the effort in chemical and programming thought needed to provide them. The drawing program, for example, must work with a standard computer terminal so that remote users can access and effectively use the program at minimal cost. It must produce unambiguous structures, even if they are not always drawn the way a chemist would like, i.e., bonds may cross, but every effort is made to prevent atom or bond overlap. The way in which complex structures are laid out in a two-dimensional template parallels some strategies used by chemists in trying to draw a structure.[10]

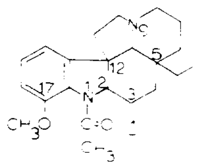
In describing the method, we have chosen to illustrate the various steps with an example. In selecting an example, we are faced with the problem which we have mentioned previously.[3b] Simple examples have the advantage of brevity but stand the risk of appearing trivial and do not begin to illustrate the power and flexibility of a program. A number of complex examples will be very tedious to wade through. The best example is one conceived and tried by the reader, and a mechanism for doing this is available (see Experimental Section). However, the traditional mechanism of presentation and sharing of results in the scientific literature is still the accepted means of dissemination of scientific information.

Table I. The Three Superatoms Used to Generate Structures for Aspidospermine (1)

Structure	Name	Restrictions
	IND	All free valences bonded to nonhydrogen atoms[a]
	ET	
	NP	All free valences bonded to nonhydrogen atoms

[a] The program also allows specific atoms within a superatom to be associated with a specified range of hydrogen atoms.

We use the structure of aspidospermine (1) as an example. As such it is somewhat artificial because it is a known structure. We add further artificiality in that we do not use the information about the structure in exactly the order in which it was determined because the history of the solution of the structure is long and is marked by several conflicting pieces of evidence which were resolved only later. We make no attempt to do the problem most efficiently with the program; our goal is primarily to illustrate several different aspects of our method with a molecule of this alkaloid's complexity.[11]



Structural Information. The earliest information recognized a dihydroindole nucleus possessing an N-acetyl group and an 8(now C-17)-methoxyl substituent on the aromatic ring.[12] Although Chalmers et al.[12c] made no assumptions as to the substitution pattern on carbons 2 and 3 (now C-2 and C-12, respectively), Witkop and Patrick[12b] assumed that C-3 was quaternary and C-2 was tertiary from chemical studies and' apparently by analogy to the structure of strychnine. Using the latter assumption, we take as our first superatom structure **2** (named arbitrarily IND, Table I), where bonds with an unspecified terminus are called "free valences"[3b]. In this case, the program is also supplied with the restriction that these free valences must be bonded to nonhydrogen atoms (although in general a free valence may be connected to any type of atom). If no assumption is made as to the number of hydrogens on carbons 2 and 3, the number of possible structures would be much larger, of course. An ethyl group was also recognized at this point. This provides our second superatom, structure **3** (named ET, Table I). This information was insufficient to solve the structure (yet both groups,[12] reasoning by analogy and chemical results proposed a structure which was nearly correct).

Additional information was needed to reduce the problem, and NMR studies of Conroy et al.[13a] provided much additional information. Even so, without further simplifica-

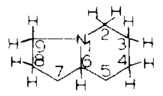
Table II. Constraints Which Are Currently Available in CONGEN[a]

Constraint type	Contains entries of the form
BADLIST	Names of undesired substructures
GOODLIST	Names and numbers of desired substructures
BADRINGS	Integer numbers representing undesired ring sizes
GOODRINGS	Integer numbers representing desired ring sizes and the number of rings of each size
PROTON	Names of substructures which define the environment of hydrogen atoms and the corresponding numbers of such atoms
ISOPRENE	Number of isoprene units and their linkage

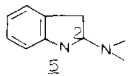
[a] There are ways of limiting the types of structures which are generated which are not formally included on the CONSTRAINTS list. We have previously mentioned the association of hydrogen atoms (may be none) with atoms bearing free valences in superatoms. This can also be done with substructures used as constraints. In addition, the program can be restricted to generate only structures with all atoms in a single system of rings.

tion, the problem was still too complex. We have worked this problem part way through based on the information of Conroy et al.,[13a] using constraints similar to those discussed below, and know that there are over 5000 intermediate structures at the first level (Figure 1) compared with only 59 in the example below.

Subsequent chemical degradation work by Conroy et al.[13b] placed the second nitrogen atom in the substructure **4.** Our program requires nonoverlapping superatoms. Carbons 5 and 7 of **4** could be carbons of the indole superatom **2** so they cannot be rigorously placed in another superatom. Carbon 6 of **4** cannot be the indole C-2 because earlier work[12b] eliminated an eserine relationship between the nitrogen atoms (i.e., **5**). This information allows us to define a third superatom (**6,** named NP in Table I) comprising carbon atoms 1-4, 6, 8, and 9 of **4.** We again add the restriction that all free valences of **6** must be bonded to nonhydrogen atoms.



4



5

We define for the program superatoms **2, 3** and **6** (Table I), using a structure editor we call EDITSTRUC.[14] All information on structural fragments (excepting atoms), whether used to construct structures or as constraints, is supplied to the program via this structure editor, which is based on concepts developed by Feldmann[15] with extensions necessary for application to our problem.

Composition and Constraints. Every structure generation problem handled by CONGEN has two basic elements: (1) a COMPOSITION list of superatoms and/or individual atoms from which structures must be generated, and (2) a CONSTRAINTS list of information which will constrain the generation procedure. These lists are utilized by a user in the following way.

COMPOSITION List. The COMPOSITION list may contain superatoms (e.g., Table I) and standard chemical atoms. On this list are placed the names of each atom or superatom and the number of each type. (The combined total of the atoms and atoms in superatoms makes up the empirical formula of the unknown compound.)

In our example, we have superatoms (one each) IND, ET, and NP (Table I), together with three remaining carbon atoms and three degrees of unsaturation (rings plus multiple bonds) to make up the empirical formula of aspidospermine ($C_{22}H_{30}N_2O_2$). We supplied this information to the program as the COMPOSITION list.

Table III. CONSTRAINTS List Used for Generation of Intermediate Structures of Aspidospermine (1)

Constraint type	Entries[a]
BADRINGS	3
GOODRINGS	2 (exactly 4)[b]
BADLIST	MET (7)[b]
	BL1 (8)

[a] "Entries" refers to responses given by the user to requests from CONGEN. They are interpreted as summarized in Table II. [b] See text for explanation.

With this list specified, we could instruct the program to GENERATE intermediate structures (Figure 1). However, in this problem as in most problems, there are constraints which are limitations to the types of structures which are plausible. Many problems quickly become unmanageably large without such limitations.[16] Our example yields 255 intermediate structures without constraints, rather than the 59 structures from GENERATE utilizing constraints as discussed below.

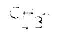
CONSTRAINTS List. Constraints which are currently available in CONGEN are summarized in Table II (not all will be used in the example). Substructures used as CONSTRAINTS types such as BADLIST, GOODLIST, and PROTON (Table II) are defined using the structure editor EDITSTRUC.

The program begins initially with no constraints; e.g., there are no internal rules of chemical feasibility. The user has the capability for defining, saving, and retrieving commonly used constraints, e.g., a general chemical BADLIST containing implausible functional groups, severely strained ring systems, and so forth. The CONSTRAINTS list can be modified at any time during a problem at the discretion of the user.

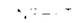
In our example, there are several constraints available from spectroscopic and chemical studies.[12,13] For the initial GENERATE to obtain intermediate structures (Figure 1), we use the constraints summarized in Table III.

There is no evidence (e.g., NMR) for three-membered rings of any type so BADRINGS was specified as three. The program regards multiple bonds as "two-membered" rings. Within the superatom IND (**2**), there exist four two-membered rings. There is evidence that there are no additional multiple bonds.[12] Thus, on GOODRINGS, we include the information that there be exactly four such rings. This prevents generation of new multiple bonds. Both kinds of ring constraints are taken by CONGEN to be statements about final structures. However, BADRINGS and GOODRINGS constraints are used at all levels within CONGEN to limit possible structures.

BADLIST contains entries which are *names* of undesired substructures (Table II). MET and BL1 are defined (using EDITSTRUC) to have structures 7 and **8,** respectively. BADLIST (and GOODLIST and PROTON) are taken by CONGEN to be statements about the structures which will be produced by the current operation, e.g., intermediate structures following a GENERATE (see below) or structures which exist in memory. Thus, **7** prevents construction of any additional methyl groups because the methyl groups in IND and ET are not "visible" to CONGEN until later imbedding (see below) takes place. Because of the known substructure **4,** the ethyl group ET cannot be bonded to NP (remember that NP (Table I) does not contain atoms 5 and 7 of **4,** which might be bonded to ET). As discussed previously intermediate structures will contain superatom names, NP,



7



8

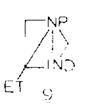
IND, and ET in our example. Thus, substructure **8** on BADLIST forbids intermediate structures in which the "atoms" NP and ET are bonded.

GENERATE Intermediate Structures. In the GENERATE phase of the program, structures are constructed from the names entered on the COMPOSITION list, using the structure generator[3b,c] within CONGEN. If the COMPOSITION list contained only atoms, then final structures result. For most problems, however, this list also contains superatom names. Each such name appears to the program at this point as a single atom possessing a valence equal to the free valence of the corresponding superatom. Because the problem is reduced to one involving only "atoms", the method of exhaustive irredundant structure generation described in our earlier work[3b,c] can be used. In the CONGEN system, this method is automatically constrained by the CONSTRAINTS list.

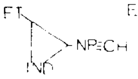
The GENERATE function employs several strategies for implementation of constraints. These strategies were developed through our discussions of typical structure elucidation problems with other chemists to elicit common heuristics ("rules of good play"). These strategies were then folded into the mechanism for structure generation. As an example, the systematic treatment of structure generation begins by allocating the atoms (and now also superatoms) in an empirical formula in all ways between groups of atoms used to form one or more ring systems and groups used to form acyclic chains (the "superatom partitions"[3b]). Each partition is analyzed in turn. If any subpart of a partition is disallowed by available constraints, then no legal structures can be constructed from that partition, and it is discarded without proceeding further. Also, it is clearly most efficient to test the smallest subpart of each partition first in the hopes that it will be illegal. Therefore, an estimator of the size of each subpart of a partition is used to select the order in which the subparts are tested. The parallel with a person's strategy may be seen from the following example. If one way of constructing structures involves a subproblem of two carbon atoms and one unsaturation, then only C=C can be built from this subproblem (to be linked in some way to the rest of the structure). If no C=C's are desired (or no additional two-membered rings in our example), then all ways of building structures using C=C can be discarded, independent of the size or any other characteristics of the rest of the problem. Another strategy is to build all tree structures[3b] from each partition before constructing ring systems from each of the subparts of a partition. There may be acyclic constraints (e.g., no additional methyl groups in our example) which can be tested in this way independent of the characteristics of the ring systems.

Specification of ranges of hydrogen atoms (as in our example, see Table I) on atoms which bear free valences in superatoms implies additional constraints which are implemented automatically at this level. For example, tests are made which ensure that, in every intermediate structure, NP is connected to four other nonhydrogen elements of the structure.

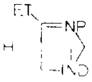
In our example, the GENERATE command, using the COMPOSITION and CONSTRAINTS lists defined previously, yields 59 intermediate structures, of which **9–13** are representative examples. Note that **9–13** have the proper composition and satisfy the specified constraints. Although **9** may appear to violate the BADRINGS = 3 constraint, in fact the program has determined that when the superatoms NP and IND are expanded to their full identities, "legal" ring sizes can result.



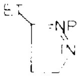
9



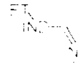
10



11



12



13

Table IV. CONSTRAINTS List Used during Imbedding NP

Constraint type	Entries
BADRINGS	3
GOODRINGS	2 (exactly 4)
	5 (at least 2)
	6 (at least 2)

Although a user well knows the internal structure of a superatom which appears in intermediate structures, such as **9–13**, CONGEN, at this point, has not associated the internal structure with the name of the superatom appearing in intermediate structures. It is the function of IMBED (see below) to expand the superatoms to their full identities and to specify the final connections among atoms. Note that the correct structure for aspidospermine will be obtained from **12.**

If examination by the chemist of these 59 intermediate structures revealed undesired features, they could be eliminated using the PRUNE command (see below) with additional constraints. In this case, no such features were seen so no pruning was done at this point.

IMBED NP. Subsequent to generation of intermediate structures, a stepwise process of imbedding the superatoms in the intermediate structures is begun (Figure 1). At each step, existing or new constraints can be brought to bear to discard unwanted structures. These procedures are outlined below.

The function of IMBED is to restore each superatom name in intermediate structures back to its original structure. This procedure can be visualized as picking up the structure of a specified superatom and substituting it in place of its name in each intermediate structure, at the same time connecting atoms in the superatom to atoms in the intermediate structure. Simultaneously, it is possible for the imbedder to form a number (user specified) of new bonds within superatoms, although no such bonds are required in our example (see APPLICATIONS).

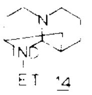
This procedure is done with cognizance of the symmetries of both the superatom and the intermediate structure. The recognition of symmetry prevents blind interconnections of atoms, which in many cases yields vast numbers of duplicate structures. The method[17] completely avoids duplication within its mathematical context. However, in our chemical context, duplicates can arise from any imbedding which increases the symmetry of the intermediate structure. Therefore, the portion of the program which does the imbedding converts each structure to a canonical form (related to that earlier described[6c]), and subsequent tests performed on this form efficiently remove what duplicates were constructed.

In general the symmetries of the superatoms and the intermediate structures in which they are imbedded are such that several (perhaps hundreds for larger cases) of unique imbeddings may result from each intermediate structure. The user has the option of specifying CONSTRAINTS to discard undesired structures as they are obtained from imbedding. When the user exercises this option, the program does not have to store large numbers of structures for subsequent pruning.

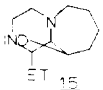
For our example, we choose (arbitrarily) to IMBED NP (**6,** Table I) first. We use the constraints summarized in Table IV during this imbedding.

We no longer need to test for the BADLIST entries in Table III. The new entries (5 and 6) on GOODRINGS include the known[12] five- and six-membered rings in the indole superatom IND and the five- and six-membered rings in the known[13b] substructure **4** which must be present in final structures. We do not necessarily want exactly two of each ring size; thus the specification of "at least 2" in Table IV. Note that these constraints are not sufficient to guarantee the presence of **4** after imbedding. We have intentionally omitted a test for the presence of **4** to illustrate the use of the PRUNE function (see below).

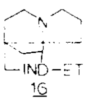
A total of 196 structures results from this IMBED. Representative structures include **14–18.** Note that the atom named NP no longer appears but has been replaced by its corresponding structure. Superatoms IND and ET remain to be imbedded.



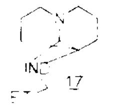
14



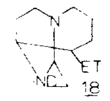
15



16



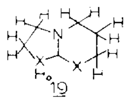
17



18

PRUNE. The PRUNE command is used to discard undesired structures from the current group of intermediate or final structures in memory. Pruning is done using constraints. As mentioned above, pruning may be done automatically during GENERATE and IMBED. It can also be done independently to apply a new constraint to an existing group of structures. The capability for pruning based on new ideas or data is very useful. The current status of a problem can be saved while new data on the unknown are collected (perhaps using experiments suggested by the alternative structural possibilities). The structures can then be retrieved and pruned on the basis of the new information.

In our example, we may notice by examining structures **14–18** that they do not contain substructure **4,** nor will they on subsequent imbeddings. Thus, to reduce the problem at this point, prior to further imbeddings which may yield many more structures, we will implement a PRUNE test to reflect our knowledge of **4.** To perform this test, we define (using EDITSTRUC) substructure **19,** called GL1. Note that the atoms numbered 5 and 7 in GL1 (**19**) are given atom names "X". X is used by convention in CONGEN to mean any nonhydrogen atom or superatom name. Atoms 5 and 7 cannot be specified only as carbon atoms at this point because the superatom IND (**2**) still exists as a single "atom" in the intermediate structures, e.g., **14–18.** Also supplied with **19** are the indicated hydrogen atom distributions as these are known (see **4**). Atom 7 of **19** (see **4**) must have no hydrogens (H_0). Atom 5 has no hydrogen range specified so may have any number of hydrogens.

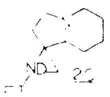


19

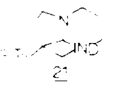
The command PRUNE to CONGEN is issued using GL1 (**19**) as a GOODLIST entry. We specify at least one GL1 as we wish to test only for its presence. The pruning is remarkable as only ten structures survive this test (**20–29**). The correct structure (**1**) will be obtained from **23.**

Table V. CONSTRAINTS List Used during Imbedding of IND (2)

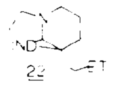
Constraint type	Entries
BADRINGS	3
GOODRINGS	2 (exactly 4)
	5 (at least 2)
	6 (at least 2)
GOODLIST	GL2 (4) (at least 1)



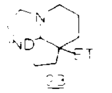
20



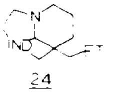
21



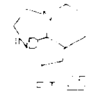
22



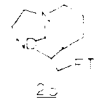
23



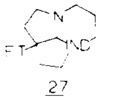
24



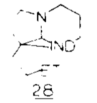
25



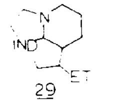
26



27



28

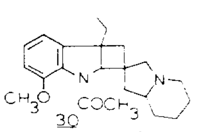


29

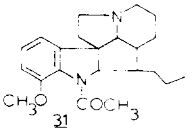
IMBED and PRUNE to Final Structures. IMBED ET. Imbedding superatom ET is done without any constraints. Only one structure results from each imbedding; this is a special case as there is only one way to imbed a monovalent superatom (and only one way to imbed an *n*-valent superatom with all free valences on one atom).

IMBED IND. Superatom IND is the only remaining superatom to be imbedded (note that C-7 of **4** is known to bear no hydrogen atoms[13b]). During this IMBED, we use the CONSTRAINTS summarized in Table V.

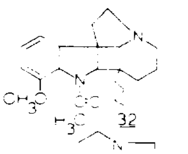
The entry GL2 on GOODLIST (Table V) is the substructure **4,** which can now be represented explicitly as **4** as no other superatoms remain to be imbedded. Eleven structures result from this IMBED, **30–40.** These are final structures.



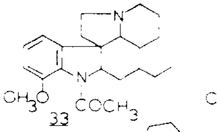
30



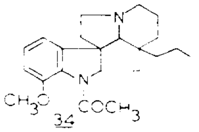
31



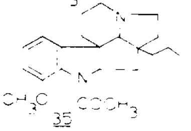
32



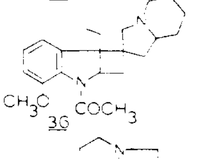
33



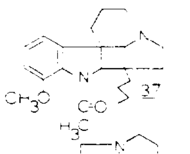
34



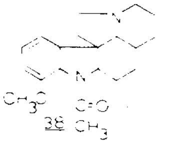
35



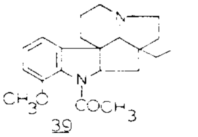
36



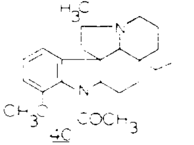
37



38

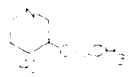


39



40

PRUNE. **30–40** represent a variety of structural types including structures with four-membered rings, spiro centers, and propyl and butyl groups. If the evidence for an ethyl group[12,13a] is considered to exclude rigorously propyl and butyl groups, then only six structures remain, **30, 33, 36,** and **38–40.** Note that if Witkop's original chemical degradation work,[12b] which isolated 3,5-diethypyridine (among other compounds), were interpreted to mean that structures must possess at least one ethyl group on a six-membered ring containing the nonindole nitrogen atom, i.e., substructure **41,** then pruning with **41** on GOODLIST (or at this point, manual examination) yields only one structure (**40**) the correct structure of aspidospermine (**1**).



41

Summary of Method

CONGEN has been structured to permit flexibility in using the program to solve a given problem. Although the most efficient strategy is to use as much information as early in the problem as possible, there is no need to do so unless, by failure to use sufficient constraints, one exhausts available storage capacity of the program. This means that one can do a GENERATE without constraints, followed by stepwise use of PRUNE to determine the effects of each constraint on reducing the scope of the problem. Alternatively, constraints can be used so that the results of GENERATE have been pruned with all available information. IMBED can be used for any chosen superatom, together with use of no constraints, a prespecified list of constraints, or a temporary list for the particular imbedding. This allows the user to select more efficient ways of arriving at his final structures by juggling imbeddings and constraints.

Storage capacity of CONGEN is finite. Although it is easily possible to save thousands of final structures on a file as they are generated (albeit very inefficiently), our goal has been to avoid strenuously this approach except for special cases where compendia of structures are required. Our reasoning is that most chemists are very unhappy with the knowledge that, under given constraints, there are still hundreds of possibilities. Although this number may serve to quantify why the structure is not yet solved, most chemists find it pointless to browse through many possibilities except to gain ideas on what additional experiments might serve to reduce the problem. However, the program must be capable of handling large problems (in terms of potential numbers of possibilities) which are also heavily constrained so that the number of final structures is manageable. Thus, our efforts have been directed much more vigorously to implementation of constraints as early and as efficiently as possible, rather than to storing away thousands of structures for retrospective pruning. The goal is clearly to avoid construction of unwanted structures at all costs. We feel that the current program is capable of handling problems where the COMPOSITION list of atoms and superatoms could conceivably yield millions of structures, but where CONSTRAINTS are sufficiently restrictive that the number of final results is small and can be arrived at without using unreasonable amounts of computer time.

Applications

In this section, we briefly outline some areas of application of CONGEN to various molecular structure problems and give an example of a recent use of the program in each area.

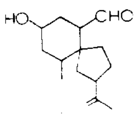
Substitution Isomers. Determination of the possible substitution products of a set of ligands about a given molecular skeleton is possible using the constructive graph labeling[3c] features of the program. We have determined the structures of different types of polychlorinated hydrocarbons in this way.[18] These problems can be done exhaustively or constrained at the user's discretion.

Ring Systems. The ability to construct structures with all atoms in a single system of rings (using a command RINGGEN rather than GENERATE) allows a user to explore questions concerning various types of ring systems, with or without contraints on plausible structures. We have outlined the method and some results obtained using CONGEN in this way.[1]

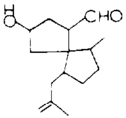
Terpene Skeletons. We are currently exploring the scope of structural isomerism of terpenoid skeletons to determine the implications of the existence of only a few representatives of such isomers out of the many possible.[19] The CONSTRAINT type ISOPRENE provides an efficient way of restricting CONGEN to construct only structures which obey some form of the isoprene rule. The particular type of linkage (e.g., head-to-tail or "any") among isoprene units can be specified by the user.

Natural Products. We are using CONGEN in the study of the structure of an unknown sesquiterpene hydrocarbon from a marine source. This problem is particularly interesting as it involves a large, ten-valent superatom which may have zero, one, or two bonds interconnecting free valences within the superatom. The three cases yield, respectively, 13, 3, and 0 intermediate structures from the initial GENERATE with constraints. The CONSTRAINTS list completely eliminated one whole segment of the problem at the first level (Figure 1). The remaining two cases yield, respectively, 148 and 31 final structures after all imbeddings. This is still a large number considering that a great deal of spectroscopic and chemical data are available on the molecule. Another interesting aspect of this problem is that the complete "degree sequence" (the number of atoms of each degree) is available from ^{13}C NMR studies. The 179 final structures were tested using PRUNE with this information, and *no* structures were eliminated. The combination of other constraints implicitly resulted in structures which all possess the same number of carbons atoms of the same degree. We suspect that similar instances of redundant information will be encountered in other problems. Subsequent pruning using substructures based on analogy with other, known compounds isolated from the same source yields a set of four "most plausible" structures. We will discuss this problem separately when the structure is solved.

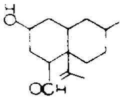
Stoessl, et al., have recently proposed a new structure for lubimin.[20] They have inferred, from ^{13}C and proton NMR spectra, a number of structural features of lubimin. They have proposed the structure **42** on the basis that it is the only possibility which possesses a known, naturally occurring skeleton. Using CONGEN, we have determined that there are 206 structures which obey the reported constraints. Inclusion of the constraint that there be no cyclopropyl hydrogens (which would probably have been visible in the proton NMR spectrum) reduces this number to 123. We have verified that no other structures besides **42** are based on a bicyclic terpenoid skeleton reported by Devon and Scott.[21] However, most of the remaining 122 structures are chemically (if not biogeneticlly) plausible, e.g., **43, 44.** Note that **42** does not obey the isoprene rule. Using the CONSTRAINT type ISOPRENE, we have determined that, of the 123 structures (above), 12 obey the isoprene rule independent of the linkage of isoprene units. Of these, only two obey a head-to-tail linkage (e.g., **45**) but are chemically less plausible than **43** or **44.**



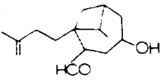
42



43



44



45

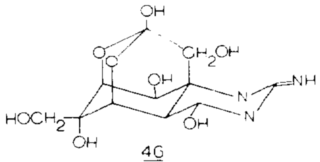
Urinary Metabolites. We are currently using CONGEN together with manual inferences derived from examination of mass spectra of unknown compounds observed in extracts of human urine. The unknowns apparently represent classes of compounds which have not previously been characterized by mass spectrometry nor exist in current libraries of mass spectra. Constraints have come from characteristic fragment ions in these spectra and from knowledge of the chemical pretreatment of the sample.

Limitations

The present program has some significant limitations which we discuss in this section. The two important limitations are problem size and representation.

Problem Size. There is currently no way to predict the scope of a structure generation problem under constraints.

Our own intuition fails us badly, distressingly often. A recent test problem using information available early in the course of the elucidation of the structure of tetrodotoxin (**46**)[22] involved a large (13-atom) superatom (with many free valences), six hydroxyl superatoms, and a very small collection of carbons, oxygens and unsaturations left over. Nearly 400 intermediate structures were built (under fairly severe constraints). We have a very crude estimate that imbedding of the large superatom (under these constraints) in all intermediate structures will yield a total of 2×10^8 final structures, a completely unforseen result. It does not matter if the estimate is in error by five orders of magnitude; the problem is still too large. We are working on ways to measure the current progress of CONGEN as it proceeds through a problem to warn the user that the problem seems too large. In these cases, as in manual efforts, the solution is to gather more information. Frequently, placement of a single additional atom in a superatom reduces a problem from huge to reasonable. This statement is probably obvious to anyone who has worked on solving a structure, but we can quantify this statement now, and the reductions can be astounding.



46

Representation. CONGEN presently produces no information on stereochemistry or any other three-dimensional property of molecules; all structures are topological representations of chemical structure.[3b]

The stereoisomeric properties of a molecule are critical data for the complete elucidation of its structure. However, these data are not usually required to determine the topological structure of an unknown. The topological structure is an important milestone in elucidation of a structure which may exist in one or more stereoisomeric forms. An algorithm for calculation and representation of such stereoisomeric properties has recently been published[6e] and could be used in CONGEN to summarize the potential stereoisomers of each candidate structure.

Conclusions

We have briefly described an approach to computer assisted structure elucidation. CONGEN provides the capability of ensuring that no plausible alternatives have been overlooked. Tentative assignment of a structure to an unknown compound can be done with much more confidence if all other candidates have been rejected using plausible constraints.

We have avoided discussion of many aspects of CONGEN, such as programming strategies, data structures, constraint implementation, and the CONSTRAINT type PROTON, to name just a few. Complex programs like CONGEN are procedures which are related to other chemical procedures in synthesis and analysis. But whereas the chemical procedures may involve a few, at most a few dozen steps, CONGEN involves hundreds of thousands of procedural steps for a typical problem. Such procedures defy description and severely strain the capabilities and usefulness of the scientific journals for their presentation. We hope that the opportunity mentioned in the Experimental Section helps bridge the gap between this brief description of CONGEN and a deeper understanding of the method and its potential applications.

Experimental Section

The parts of CONGEN which deal with structure generation and the user interface are written in the program language INTERLISP. Imbedding and ancillary canonicalization and pruning while imbedding are implemented in SAIL. The structure drawing program is written in FORTRAN. The program runs on a Digital Equipment Corporation KI-10 computer at the Stanford University Medical Experimental (SUMEX) computer facility. This facility was established to promote sharing of such complex programs, which would be difficult to mount on another computer system. CONGEN is available to an outside community of users (to the limits of available resources) via a nationwide computer network. The program can be accessed over standard telephone lines using any of a variety of computer terminals. From many cities, this represents only a local telephone call.

A copy of the CONGEN documentation is available to interested persons. It provides more detailed information on the method and use of the program. Those interested in gaining access to the program or learning more about other facilities of SUMEX should write to the authors or to Professor J. Lederberg, Principal Investigator, SUMEX Project, Stanford University Medical School, Stanford, Calif. 94305.

References and Notes

[1] For Part XVI, see R. E. Carhart, D. H. Smith, H. Brown, and N. S. Sridharan, *J. Chem. Inf. Comput. Sci.*, **15**, 124 (1975).

[2] This work was supported by the National Institutes of Health, Grants RR 00612-05A1 and RR 785-01A1; the latter in support of the Stanford University Medical Experimental Computer Facility, SUMEX.

[3] (a) D. H. Smith, B. G. Buchanan, R. S. Engelmore, A. M. Duffield, A. Yeo, E. A. Feigenbaum, J. Lederberg, and C. Djerassi, *J. Am. Chem. Soc.*, **94**, 5962 (1972); (b) L. M. Masinter, N. S. Sridharan, J. Lederberg, and D. H. Smith, *ibid.*, **96**, 7702 (1974); (c) L. M. Masinter, N. S. Sridharan, R. E. Carhart, and D. H. Smith, *ibid.*, **96**, 7714 (1974).

[4] By "artificial intelligence" programs, we refer to programs for semantic information processing which are designed to emulate human performance in problem-solving activities. As far as possible (but not necessarily), we attempt to model in the program the reasoning processes of chemists in proceeding from data to conclusions based on those data. For a discussion of this area of computer science research, see (a) M. Minsky, Ed., "Semantic Information Processing", MIT Press, Cambridge, Mass., 1968; (b) Schank and K. Colby, Ed., "Computer Models of Thought and Language", W. H. Freeman, San Francisco, Calif., 1973.

[5] N. Nilsson, "Problem Solving Methods in Artificial Intelligence", McGraw-Hill, New York, N.Y., 1971.

[6] (a) E. J. Corey and W. T. Wipke, *Science*, **166**, 178 (1969); (b) E. J. Corey, W. T. Wipke, R. D. Cramer, III, and W. J. Howe, *J. Am. Chem. Soc.*, **94**, 421 (1972); (c) *ibid.*, **94**, 431 (1972); (d) E. J. Corey, R. D. Cramer, III, and W. J. Howe, *ibid.*, **94**, 440 (1972); (e) W. T. Wipke and T. M. Dyott, *ibid.*, **96**, 4825 (1974); (f) H. Gelernter, N. S. Sridharan, A. J. Hart, S.-C. Yen, F. W. Fowler, and H. J. Shue, *Fortschr. Chem. Forsch.*, **41**, 113 (1973); (g) E. J. Corey, W. J. Howe, and D. Pensak, *J. Am. Chem. Soc.*, **96**, 7724 (1974).

[7] (a) S. Sasaki, H. Abe, T. Oubi, M. Sakamoto, and S. Ochiai, *Anal. Chem.*, **40**, 2220 (1968); (b) S. Sasaki, Y. Kudo, S. Ochiai, and H. Abe, *Mikrochim. Acta*, 726 (1971); (c) Y. Kudo, *Kagaku No Ryoiki, Zokan*, **98**, 115 (1972); (d) Y. Kudo, personal communication.

[8] (a) M. E. Munk, C. S. Sodano, R. L. McLean, and T. H. Haskell, *J. Am. Chem. Soc.*, **89**, 4158 (1967); (b) D. B. Nelson, M. E. Munk, K. B. Gash, and D. L. Herald, Jr., *J. Org. Chem.*, **34**, 3800 (1969); (c) B. D. Cox, Ph.D. Thesis, Department of Chemistry, Arizona State University, 1973.

[9] L. A. Gribov, V. A. Demontyer, M. E. Elyashberg, and E. Z. Yakupov, *J. Mol. Struct.*, **22**, 161 (1974).

[10] R. E. Carhart, unpublished results.

[11] (a) L. Marion in "The Alkaloids", Vol. II, R. H. F. Manske and H. L. Holmes, Ed., Academic Press, New York, N.Y., 1952, p 369; (b) J. E. Saxton in *ibid.*, Vol. VII, R. H. F. Manske, Ed., 1960, p 4; (c) B. Gilbert in *ibid.*, Vol. VIII, 1965, p 336.

[12] (a) B. Witkop, *J. Am. Chem. Soc.*, **70**, 3712 (1948); (b) B. Witkop and J. B. Patrick, *ibid.*, **76**, 5603 (1954); (c) J. R. Chalmers, H. T. Openshaw, and G. F. Smith, *J. Chem. Soc.*, 1115 (1957); (d) A. J. Everett, H. T. Openshaw, and G. F. Smith, *ibid.*, 1120 (1957).

[13] (a) H. Conroy, P. R. Brook, M. K. Rout, and N. Silverman, *J. Am. Chem. Soc.*, **80**, 5178 (1958); (b) H. Conroy, P. R. Brook, and Y. Amiel, *Tetrahedron Lett.*, **11**, 4 (1959).

[14] We adopt the convention of using small capital letters to designate various commands used within CONGEN, e.g., EDITSTRUC, GENERATE. The names of these commands are also descriptive of their function and so are frequently used in the text as adjectives or verbs.

[15] R. Feldmann in "Computer Representation and Manipulation of Chemical Information", W. T. Wipke, S. R. Heller, R. J. Feldmann, and E. Hyde, Ed., Wiley, New York, N.Y., 1974, p 55.

[16] D. H. Smith, *Tetrahedron, J. Chem. Inf. Comput. Sci.*, in press.

[17] H. Brown, *SIAM J. Comput.*, in press.

[18] D. H. Smith, *Anal. Chem.*, **47**, 1176 (1975).

[19] D. H. Smith and R. E. Carhart, unpublished results.

[20] A. Stoessl, J. B. Stothers, and E. W. B. Ward, *J. Chem. Soc., Chem. Commun.*, 709 (1974).

[21] T. K. Devon and A. I. Scott, "Handbook of Naturally Occurring Compounds", Vol. II, Academic Press, New York, N.Y., 1972.

[22] (a) T. Goto, Y. Kishi, S. Takahashi, and Y. Hirata, *Tetrahedron Lett.*, 779 (1964); (b) R. B. Woodward and J. Z. Gougoutas, *J. Am. Chem. Soc.*, **86**, 5030 (1964).